Research Proposal

Effective and Efficient Clustering of High Dimensional Data

Abstract: Data mining focuses on finding previously unknown yet potentially useful, hidden patterns from large amounts of data. Clustering is one of the most commonly used unsupervised data mining techniques, and it has been successfully applied to find groups of similar data points in many applications. However, conventional clustering algorithms sometimes fail to find meaningful clusters when dataset has dozens of attributes because the high dimensionality makes the data space very noisy.

Clustering real-world data sets is often hampered by the so-called curse of dimensionality: many real-world data sets consist of a very high dimensional feature space. In general, most of the common algorithms fail to generate meaningful results because of the inherent sparsity of the data space. Usually, clusters cannot be found in the original feature space because several features may be irrelevant for clustering due to correlation and/or redundancy. However, clusters are usually embedded in lower dimensional subspaces. In addition, different sets of features may be relevant for different sets of objects. Thus, objects can often be clustered differently in varying subspaces of the original feature space. To over come from these difficulties, we will define and develop efficient and effective approaches for clustering large multi dimensional data.

Background: With an increasing number of new database applications dealing with very large high dimensional data sets, data mining on such data sets has emerged as an important research area. These applications include multimedia content-based retrieval, geographic and molecular biology data analysis, text mining, bioinformatics, medical applications, and time-series matching. In many of the above applications the data sets are very large, consisting of millions of data objects with several hundreds to thousands of dimensions.Clustering techniques are categorized into several different approaches: partitioning, hierarchical, density based, subspace based, grid based, model based and soft computing. Given a data set with n objects and $k \leq n$, the number of desired clusters, *partitioning algorithms* partition the objects into k clusters. The clusters are formed in order to optimize an objective criterion such as distance. Each object is assigned to the closest cluster. Clusters are typically represented by either the mean of the objects assigned to the cluster (k-means) or by one representative object of the cluster (k-medoid [6]). CLARANS (Clustering Large

Applications based upon RANdomized Search) [7] is a partitioning clustering algorithm developed for large data sets, which uses a randomized and bounded search strategy to improve the scalability of the *k*-medoid approach. CLARANS enables the detection of outliers and its computational complexity is about $O(n^2)$. Hierarchical clustering algorithms work by grouping data objects into a hierarchy (e.g., a tree) of clusters. The hierarchy can be formed top-down (*divisive* hierarchical methods) or bottom-up (*agglomerative* hierarchical methods). Hierarchical methods rely on a distance function to measure the similarity between clusters. These methods do not scale well with the number of data objects. Their computational complexity is usually $O(n^2)$. Some newer methods such as BIRCH [12] and CURE [4] attempt to address the scalability problem and improve the quality of clustering results for hierarchical methods. BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) is an efficient divisive hierarchical algorithm. It has $O(n)$ computational complexity, can work with limited amount of memory, and has efficient I/O. It uses a special data structure, CF tree (Cluster Feature tree) for storing summary information about sub clusters of objects. Because of the similarity measure it uses to determine the data items to be compressed, BIRCH only performs well on data sets with spherical clusters. CURE (Clustering Using RE presentatives) is an $O(n^2)$ algorithm that produces high-quality clusters in the presence of outliers, and can identify clusters of complex shapes and different sizes. It employs a hierarchical clustering approach that uses a fixed number of representative points to define a cluster instead of a single centroid or object. CURE handles large data sets through a combination of random sampling and partitioning. Since CURE uses only a random sample of the data set, it manages to achieve good scalability for large data sets. CURE reports better times than BIRCH on the same benchmark data.

Locality-based clustering algorithms group neighboring data objects into clusters based on local conditions. These algorithms allow clustering to be performed in one scan of the data set. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [10] is a typical representative of this group of algorithms. It regards clusters as dense regions of objects in the input space that are separated by regions of low density. DBSCAN's basic idea is that the density of points in a radius around each point in a cluster has to be above a certain threshold. It grows a cluster as long as, for each data point within this cluster, a neighborhood of a given radius contains at least a minimum number of points. DBSCAN has computational complexity $O(n^2)$. If a spatial index is used, the computational complexity is $O(n \log n)$. The clustering generated by DBSCAN is very sensitive to parameter choice. OPTICS (Ordering Points To Identify Clustering

In general, partitioning, hierarchical, and locality-based clustering algorithms do not scale well with the number of objects in the data set. To improve the efficiency, data summarization techniques integrated with the clustering process have been proposed. Besides the above-mentioned BIRCH and

CURE algorithms, examples include: active data clustering [5], Scalable , and simple single pass *k*-means. Active data clustering utilizes principles from sequential experimental design in order to interleave data generation and data analysis. It infers from the available data not only the grouping structure in the data, but also which data are most relevant for the clustering problem. The inferred relevance of the data is then used to control the re-sampling of the data set. Scalable-KM requires at most one scan of the data set. The method identifies data points that can be effectively compressed, data points that must be maintained in memory, and data points that can be discarded. The algorithm operates within the confines of a limited memory buffer. Unfortunately, the compression schemes used by Scalable-KM can introduce significant overhead. The simple single pass *k*-means algorithm is a simplification of Scalable-KM. Like Scalable-KM, it also uses a data buffer of fixed size.In high dimensional data sets, it is very unlikely that data points are nearer to each other than the average distance between data points because of sparsely filled space. As a result, as the dimensionality of the space increases, the difference between the distance to the nearest and the farthest neighbors of a data object goes to zero[13]. Grid-based clustering algorithms do not suffer from the nearest neighbor problem in high dimensional spaces. Examples include STING (STatistical INformation Grid) [11], CLIQUE [3], DENCLUE [8], WaveCluster [4], and MAFIA (Merging Adaptive Finite Intervals And is more than a clique) [3]. These methods divide the input space into hyper-rectangular cells, discard the low-density cells, and then combine adjacent high-density cells to form clusters. Grid-based methods are capable of discovering cluster of any shape and are also reasonably fast. However, none of these methods address how to efficiently cluster very large data sets that do not fit in memory. Furthermore, these methods also only work well with input spaces with low to moderate numbers of dimensions. As the dimensionality of the space increases, grid based methods face some serious problems. The number of cells grows exponentially and finding adjacent high-density cells to form clusters becomes prohibitively expensive.

Genetic algorithms (GAs)[14] are randomized search and optimization techniques guided by the principles of evolution and natural genetics. C.A.Murthy and N.Chowdhury [15] have considered a partition to be encoded as a string of length n. The i_{th} element of the chromosome represents the cluster number to which the corresponding point belongs. A comparison of the performance of their algorithm, subsequently referred to as the GA-clustering algorithm, with that of the K-means algorithm is also provided. S. Bandyopadhyay and U. Maulik,[16] proposed a genetic algorithm-based efficient clustering technique that utilizes the principles of K-Means algorithm. The algorithm is known KGA-clustering. It avoids major limitation of getting stuck at locally optimal values. Performance of KGA is evaluated over the K-Means algorithm and another genetic algorithm-based clustering method on several artificial and real life data sets. Gautam Garai and B.B. Chaudhuri,[17]

proposed a Genetic Clustering Algorithm is basically a two-phase process. At the first phase the original data set is decomposed into a number of fragmented clusters in order to spread the GA search process at the latter phase over the entire space. At the second phase Hierarchical Cluster Merging Algorithm [HCMA] is used. The HCMA is an iterative genetic algorithm based approach that combines some of the fragmented clusters into complete k-cluster. Sushmita Mitra,[18] proposed an evolutionary rough c-means clustering algorithm. The Davies–Bouldin clustering validity index is used as the fitness function. The relative importance of the upper and lower approximations and the threshold of the rough clusters are optimized using GAs. The clusters are modeled as c rough sets, expressed in terms of upper and lower approximations. H. J. Lin,[19] proposed a genetic clustering algorithm that select the cluster centers directly from the data set. A new point symmetry distance based genetic clustering algorithm is proposed [20],which incorporates both the Euclidean distance as well as a measure of symmetry with respect to a point in its computation. Kd-tree based nearest neighbor search is used to reduce the complexity of symmetry-based distance computation. It is able to detect arbitrary shape clusters, irrespective of their geometrical shape and overlapping nature.

Objective: Clustering large data sets of high dimensionality has always been a serious challenge for clustering algorithms. Many recently developed clustering algorithms have attempted to address either handling data sets with a very large number of records or data sets with a very high number of dimensions.Although applying full-dimensional clustering algorithms to low-to-medium-dimensional problems demonstrate good results, they usually suffer from two conditions: curse of dimensionality and decreasing simplicity with increasing dimensionality. Curse of dimensionality is a known caveat for algorithms that utilize full the attribute space when dealing with high-dimensional problems because the distance between instances become increasingly meaningless with increasing dimensionality. Simplicity, on the other hand, is a more severe issue because even if a full-attribute method can solve a problem, the solution would be too complex for end users to interpret.Therefore parsimony is motivated. A quick solution to this is utilizing feature selection methods prior to clustering, which is a common practice and has proven efficient for a large category of problems. All problems in this category, which benefit from feature selection, share the same characteristic. The second way is known as subspace clustering, which is a special class of clustering algorithms that try to circumvent high dimensionality by building clusters in appropriate subspaces of the original feature space.

Work Plan

1. Identify techniques on which clustering methods will be developed.
2. Define methods to cluster multi dimensional data sets
3. Implemet algorithmic models of clustering high dimensional data sets.
4. Quantify the predictive accuracy of the model with statistical and machine learning evaluation techniques .This will require the development of novel,robust methods for comparing existing methods.
5. Compare the results with other clustering methods

References:

[1] C.Aggarwal andP.Yu. "Finding Generalized Projected Clusters in High Dimensional Space". In Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD' 00), 2000.

[2] C. Aggarwal and C. Procopiuc. "Fast Algorithms for Projected Clustering". In Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'99), 1999.

[3] R. Agrawal, J. Gehrke, D.Gunopulos, and P.Raghavan. "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications". In Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD' 98), 1998.

[4] J. Han and M. Kamber. Data Mining: Concepts and Techniques. Academic Press, 2001.

[5] E.K.K. Ng, A.W. Fu, and R.C. Wong, "Projective Clustering by Histograms," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 3,pp. 369-383, Mar. 2005

[6] L. Kaufman, and P. J. Rousseeuw. Finding groups in data: An introduction to cluster analysis. New York: John Wiley & Sons, 1990.

[7] R. Ng, and J. Han. Efficient and effective clustering method for spatial data mining. In Proc. 1994 Int.Conf. on Very Large Data Bases (VLDB'94),pages 144–155, 1994.

[8] Hinneburg, and D. A. Keim. An efficient approach to clustering in large multimedia Databases with noise. In Proc. 1998 Int. Conf. Knowledge Discovery and Data Mining (KDD'98), pages 58–65, 1998.

[9] P. Bradely, U. Fayyad, and C. Reina. Scaling clustering algorithms to large databases. In Proc. 1998 Int. Conf. Knowledge Discovery and Data Mining (KDD'98), pages 8–15, 1998.

[10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial database. In Proc. 1996 Int. Conf. Knowledge Discovery and Data Mining (KDD'96), pages 226–231, 1996.

12/23

[11] W. Wang, J. Yang, M. Muntz. STING: A statistical information grid approach to spatial data mining. In Proc. 1997 Int. Conf. on Very Large Data Bases (VLDB'97), pages 186–195, 1997.

[12] W. Wang, J. Yang, M. Muntz. STING: A statistical information grid approach to spatial data mining. In Proc. 1997 Int. Conf. on Very Large Data Bases (VLDB'97), pages 186–195, 1997.

[13] Hinneburg, C. C. Aggarwal, D. A. Keim. What is the nearest neighbor in high dimensional spaces? In Proc. 26th Int. Conf. on Very Large Data Bases (VLDB'00), pages 506–515, 2000.

[14] Goldberg, D.E.,1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

[15] C.A.Murthy and N.Chowdhury,1996.In search of optimal clusters using genetic algorithms,pattern Recog.lett.,vol.17,pp.825-832.

[16] S. Bandyopadhyay and U. Maulik, 2002. An Evolutionary Technique Based on K-Means Algorithm for Optimal Clustering in R^N", Information Sciences, vol. 146, pp. 221-237.

[17] Gautam Garai and B.B. Chaudhuri, 2004.A novel genetic algorithm for automatic clustering, Pattern Recognition Letters, 25, pp 173–187.

[18] Sushmita Mitra, 2004. An evolutionary rough portative clustering, Pattern Recognition Letters, 25, pp 1439–1449.

[19] H. J. Lin, F. W. Yang and Y. T. Kao.2005. An efficient GA-based clustering technique, Tamkang Journal of Science and Engineering, vol. 8, no. 2,pp. 113-122.

[20] S. Bandyopadhyay and S. Saha, 2007.GAPS: A clustering method using a new point symmetry-based distance measure Technique, Pattern Recognition, vol. 40, no. 12, pp. 3430-3451.

12/07/2009

Submitted by
Vijendra Singh
Er No.:070902
Department of Computer Science & Engineering